James Crockett
2021-Nov-13
Foundations of Programming (Python)
Assignment 05

Dictionaries

Introduction

Assignment 5 covers the introduction to dictionaries and pulling data from a text file. What made this week unique is that the assignment involved updating existing code that was provided, whereas previous weeks required students to create code from scratch. One thing that was consistent was that I still had to test each code block during runtime to make sure the correct output and behavior was present.

Pseudo Code

Pseudo code was provided by default when unzipping Module 5, which organized the different code blocks required to form the final CD Inventory program.

Initial Approach

I reviewed the entire starter script for week 5 to understand how the existing code was laid out and where it differed from my week 4 script implementation. I then added the assignment instructions from the Module 5 PDF[1], Step 5.2, into a comment block after the Change Log section of the script.

Dictionaries

Dictionaries are a collection of keys and values encapsulated by curly brackets. One of the big differences with dictionaries over lists and tuples is that data is not stored in a sequence, but within pairs using keys and values. Dictionary data can be accessed directly by referencing the key name versus an index position typically used with a list or tuple. This is a useful construct to use when working with larger blocks of data.

Modifying the existing script to convert the list data into a dictionary was straightforward. I started with the 'a', Add CD entry code block, by renaming the lstRow variable name to dictRow and replacing the list brackets, [], with curly brackets, {} in line 70, Figure 1. Then I added key names to each of the former list variables separated with pairing colons to conform to syntax required of dictionaries. The next step was to go through all the variables where the lstRow variable name was used and ensure they were converted to the updated dictionary variable name and curly brackets throughout the script.

[1] Dirk Biesinger, Assignment_05.pdf, 2.

```
64      elif strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
65          # 2. Add data to the table (2d-list) each time the user wants to add data
66          strID = input('Enter an ID: ')
67          strTitle = input('Enter the CD\'s Title: ')
68          strArtist = input('Enter the Artist\'s Name: ')
69          intID = int(strID)
70          dictRow = {'ID': intID, 'Title': strTitle, 'Artist': strArtist}
71          lstTbl.append(dictRow)
```

Figure 1- list converted to dictionary construct

The objFile 'r' Access Argument

The clear() method was first used to clear out any content that may be in the 2D list table from previous user actions before reading and pulling in data from the text file into memory. To load data into memory from a text file, the "r", read argument was used for the objFile function to access the file. This provided a clean slate to start with before loading the data.

The "for in" loop was used to process each element in the text file rows into a dictionary pair outlined in line 59, which was then appended into the 2D list table, lstTbl variable, Figure 2.

```
53      if strChoice == 'l':
54          # TODO Add the functionality of loading existing data
55          lstTbl.clear()
56          objFile = open(strFileName, 'r')
57          for row in objFile:
58              lstRow = row.strip().split(',')
59              dictRow = {'ID': int(lstRow[0]), 'Title':lstRow[1], 'Artist':lstRow[2]}
60              lstTbl.append(dictRow)
61          objFile.close()
```

Figure 2-pulling in elements from text file into a dictionary

The Challenge of Deleting a row from the Dictionary

I used the "for row in lstTble" loop as a starting point but was at an impasse on how to delete the row. I attempted using an example from the Python book [2] in the section, "Deleting a Key-Value Pair. I used: del lstTbl[varKey], but this failed to delete the entire row. The delete key-value pair implementation resulted in only errors in the way I implemented the original code, probably a syntax issue on my part. I started to think of other ideas, but none panned out in providing the desired outcome. The week 5, Friday teacher office hours provided me a basic example during the Zoom session that eventually resulted in lines 95-96 in Figure 3.

I still had to test and make sure the code executed correctly. My first attempt failed when I didn't use the key name 'ID' in line 95. I used the list index [0], which failed during code execution because the lstTbl was a 2d collection of dictionaries, not lists. I thought about this error further and went back to the class recording where a student after me asked about how to convert the if row[0] == intID list example into a dictionary format, and then saw that the key name must be used instead of index

[2] Michael Dawson, Python Programming for the absolute beginner, 3rd Edition, 146

number. Once I updated line 95 with the key name, the code block execution finally passed the smoke test.

```
90          print()
91          intId = int(input('Enter the ID number you want to delete: '))
92
93          for row in lstTbl:
94              #for list, use index format: row[0], for dict, use key name ['ID']
95              if row['ID'] == intId:
96                  lstTbl.remove(row)
97                  boolDel = True #bool flag used for line 102 code block
98
```

Figure 3-deleting a row from a 2d dictionary

Summary

This week helped me build on lessons learned and showed me the advantages of joining the Friday working question and answer sessions. I implemented my new strategy of looking at the assignment the first few days of week 5 instead of working only on the labs through the week before looking at the actual assignment. This was one of the reasons why I was able to identify any blocking issues early on in implementing any parts of the assignment and field my questions in the Friday Zoom session. I was able to get more out of this week than the previous week where I was spending countless hours at a dead end with how I was trying to implement the coding objectives. Week 5 Github link is provided in the Appendix.

Appendix

The following pages in this appendix show screen captures of:

- Figure 4, Page 5 – Spyder Console code execution.
- Figure 5, Page 6 – Mac Terminal code execution.
- Github, Assignment 5 Repository: https://github.com/jamescisonline/Assignment_05

Figure 4-Spyder console code execution

```
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD and overwrite Inventory
[s] Save Inventory to file
[x] exit

l, a, i, d, s or x: l

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD and overwrite Inventory
[s] Save Inventory to file
[x] exit

l, a, i, d, s or x: d


This is your current inventory from your application memory

ID, CD Title, Artist
{'ID': 1, 'Title': 'two', 'Artist': 'three'}
{'ID': 3, 'Title': 'four', 'Artist': 'five'}
{'ID': 5, 'Title': 'six', 'Artist': 'seven'}
{'ID': 2, 'Title': 'three', 'Artist': 'four'}
{'ID': 4, 'Title': 'five', 'Artist': 'six'}
{'ID': 6, 'Title': 'seven', 'Artist': 'eight'}


Enter the ID number you want to delete: 2

This is your updated inventory. Please select "s" from main menu to save your changes.

ID, CD Title, Artist
{'ID': 1, 'Title': 'two', 'Artist': 'three'}
{'ID': 3, 'Title': 'four', 'Artist': 'five'}
{'ID': 5, 'Title': 'six', 'Artist': 'seven'}
{'ID': 4, 'Title': 'five', 'Artist': 'six'}
{'ID': 6, 'Title': 'seven', 'Artist': 'eight'}


[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD and overwrite Inventory
[s] Save Inventory to file
[x] exit

l, a, i, d, s or x: s
```

Figure 5-Mac Terminal code execution